Programação de Horários em Instituições Educacionais: Formulações e Algoritmos

SBPO 2007 - Minicurso - Parte I

Haroldo Gambini Santos e Marcone Jamilson Freitas Souza

28 de agosto de 2007

Conteúdo

Wren 1996:

"A alocação, sujeita a restrições, de recursos a objetos colocados no espaço e no tempo, de modo a satisfazer, tanto quanto possível, um conjunto de objetivos desejáveis."

	Segunda					
Turma \ Dia,Per.	1° Per.	2° Per.	3° Per.	4° Per.	5° Per.	1° Per.
101 A	Prof. João	Prof. João	Profa. Soraia	Prof. Alberto	Profa. Júlia	Prof. João
101 B	Profa. Soraia	Prof. Alberto	Prof. João	Profa. Júlia	Profa. Regina	Profa. Soraia
201 A	Profa. Júlia	Profa. Júlia	Prof. Carlos	Prof. João	Profa. Soraia	Profa. Regina
201 B	Profa. Regina	Prof. Soraia	Prof. Alberto	Prof. Carlos	Prof. Carlos	Prof. Alberto
201 C	Prof. Carlos	Prof. Carlos	Profa. Júlia	Profa. Regina	Prof. João	Prof. Carlos
301 A	Prof. Alberto	Profa. Regina	Profa. Regina	Profa. Soraia	Prof. Alberto	Prof. Celso
301 B	Prof. Celso	Prof. Celso	Prof. João	Prof. Artur	Prof. Artur	Prof. Antônio
301 C	Prof. Artur	Prof. Artur	Prof. Celso	Prof. Celso	Prof. Antônio	Profa. Ma[illegible]

Problemas de Programação de Horários

Interesse

- dificuldade de resolução
  - muitos interesses
  - solução factível
  - reaproveitamento ?
- importância prática
  - satisfação do corpo docente
  - melhor gestão de recursos
  - horário mais agradável para alunos
- importância teórica
  - simples de ser explicado, difícil de ser resolvido
  - problema de otimização com múltiplos objetivos
  - NP-Completo mesmo em casos simples

Problemas de Programação de Horários

Interesse

- dificuldade de resolução
  - muitos interesses
  - solução factível
  - reaproveitamento ?
- importância prática
  - satisfação do corpo docente
  - melhor gestão de recursos
  - horário mais agradável para alunos
- importância teórica
  - simples de ser explicado, difícil de ser resolvido
  - problema de otimização com múltiplos objetivos
  - NP-Completo mesmo em casos simples

Problemas de Programação de Horários

Interesse

- dificuldade de resolução
  - muitos interesses
  - solução factível
  - reaproveitamento ?
- importância prática
  - satisfação do corpo docente
  - melhor gestão de recursos
  - horário mais agradável para alunos
- importância teórica
  - simples de ser explicado, difícil de ser resolvido
  - problema de otimização com múltiplos objetivos
  - NP-Completo mesmo em casos simples

Dificuldades

- Definição
  - múltiplos objetivos
  - hierarquia dos objetivos ?
- Resolução
  - problemas de natureza combinatória, de difícil tratamento computacional

Dificuldades

- Definição
  - múltiplos objetivos
  - hierarquia dos objetivos ?
- Resolução
  - problemas de natureza combinatória, de difícil tratamento computacional

Definição do problema

Requerimentos

- Exemplos:
  - nenhum professor pode lecionar mais que 5 aulas em um dia
  - aulas de matemática devem ser lecionadas nos primeiros períodos
  - a carga horária semanal de um curso deve ser dividida em 2 dias da semana

Classes de Requerimentos

- organizacionais
  - gestão dos recursos da instituição:
    - salas de aula
  - cumprimento das leis trabalhistas
- pedagógicos
  - duração das aulas
  - espaçamento adequado entre as aulas
- pessoais
  - preferências por horários
  - minimização do deslocamento

Requerimentos $\longrightarrow$ Restrições

Classes de Requerimentos

- organizacionais
  - gestão dos recursos da instituição:
    - salas de aula
  - cumprimento das leis trabalhistas
- pedagógicos
  - duração das aulas
  - espaçamento adequado entre as aulas
- pessoais
  - preferências por horários
  - minimização do deslocamento

Requerimentos $\longrightarrow$ Restrições

Classes de Requerimentos

- organizacionais
  - gestão dos recursos da instituição:
    - salas de aula
  - cumprimento das leis trabalhistas
- pedagógicos
  - duração das aulas
  - espaçamento adequado entre as aulas
- pessoais
  - preferências por horários
  - minimização do deslocamento

Requerimentos $\longrightarrow$ Restrições

Classes de Requerimentos

- organizacionais
  - gestão dos recursos da instituição:
    - salas de aula
  - cumprimento das leis trabalhistas
- pedagógicos
  - duração das aulas
  - espaçamento adequado entre as aulas
- pessoais
  - preferências por horários
  - minimização do deslocamento

Requerimentos $\longrightarrow$ Restrições

Conjunto de Restrições

Particionamento

- Restrições *Fortes*
  - qualquer quadro de horários **deve** satisfazê-las
  - definem o espaço de busca
- Restrições *Fracas*
  - sua satisfação é **desejável**
  - são utilizadas para *avaliar* um quadro de horários:
    - função objetivo: minimizar a violação das restrições fracas

Conjunto de Restrições

Particionamento

- Restrições *Fortes*
  - qualquer quadro de horários **deve** satisfazê-las
  - definem o espaço de busca
- Restrições *Fracas*
  - sua satisfação é **desejável**
  - são utilizadas para *avaliar* um quadro de horários:
    - função objetivo: minimizar a violação das restrições fracas

Tratando Problemas de Programação de Horários

Abordagens Multi-Objetivo

- construir conjunto de soluções não dominadas
- decisor seleciona qual solução tem o melhor compromisso entre os objetivos, de acordo com seus critérios
  - problema: número de objetivos ≥ 3 - conjunto potencialmente muito grande de soluções não dominadas
- Ex.: M. P. Carrasco and M.V. Pato. **A multiobjective genetic algorithm for the class/teacher timetabling problem**. In Selected papers from III PATAT, volume 2079 of Lecture Notes In Computer Science, 2000.
- todas as restrições fracas agrupadas em apenas duas

Abordagens Multi-Objetivo

- construir conjunto de soluções não dominadas
- decisor seleciona qual solução tem o melhor compromisso entre os objetivos, de acordo com seus critérios
    - problema: número de objetivos ≥ 3 - conjunto potencialmente muito grande de soluções não dominadas
- Ex.: M. P. Carrasco and M.V. Pato. **A multiobjective genetic algorithm for the class/teacher timetabling problem**. In Selected papers from III PATAT, volume 2079 of Lecture Notes In Computer Science, 2000.
- todas as restrições fracas agrupadas em apenas duas

Tratando Problemas de Programação de Horários

Abordagens Multi-Objetivo

- construir conjunto de soluções não dominadas
- decisor seleciona qual solução tem o melhor compromisso entre os objetivos, de acordo com seus critérios
  - problema: número de objetivos ≥ 3 - conjunto potencialmente muito grande de soluções não dominadas
- Ex.: M. P. Carrasco and M.V. Pato. **A multiobjective genetic algorithm for the class/teacher timetabling problem**. In Selected papers from III PATAT, volume 2079 of Lecture Notes In Computer Science, 2000.
- todas as restrições fracas agrupadas em apenas duas

Combinando objetivos

- abordagem mais usual:
  - objetivos são combinados em uma única função de avaliação, onde cada objetivo recebe um peso

$$f(x) = \sum_{k=1}^{p} w_k \cdot f_k(x)$$

$$w_k \in \mathbb{R}^+$$

Problemas de Programação de Horários em Instituições Educacionais

- Programação de Horários em Universidades
  - Programação de Cursos
  - Programação de Exames
- Problema de Alocação de Salas
- Problema de Programação de Horários em Escolas

- Primeira Edição - 2002
  - Programação de Cursos em Universidades
  - Publicações subseqüentes ao concurso utilizaram as instâncias
- Segunda Edição - em curso !
  - Três categorias
    - Programação de Exames
    - Programação de Cursos em Universidades - Pós Matrícula
    - Programação de Cursos em Universidades - Baseada em Currículos

- Primeira Edição - 2002
  - Programação de Cursos em Universidades
  - Publicações subseqüentes ao concurso utilizaram as instâncias
- Segunda Edição - em curso !
  - Três categorias
    - Programação de Exames
    - Programação de Cursos em Universidades - Pós Matrícula
    - Programação de Cursos em Universidades - Baseada em Currículos

O Problema de Programação de Cursos em Universidades

O Problema na *II International Timetabling Competition*

- duas categorias:
  - programação de cursos **pós-matrícula** (*post-enrollment*):
    - alunos matriculam-se livremente em cursos, sem conhecer os seus horários;
    - horário é posteriormente construído, visando a máxima compatibilidade possível com as escolhas dos alunos.
  - programação de cursos baseada em **currículo** (*curriculum based*).
    - dois cursos são ditos conflitantes se pertencem a um mesmo currículo.

O Problema na *II International Timetabling Competition*

- duas categorias:
  - programação de cursos **pós-matrícula** (*post-enrollment*):
    - alunos matriculam-se livremente em cursos, sem conhecer os seus horários;
    - horário é posteriormente construído, visando a máxima compatibilidade possível com as escolhas dos alunos.
  - programação de cursos baseada em **currículo** (*curriculum based*).
    - dois cursos são ditos conflitantes se pertencem a um mesmo currículo.

Dados de Entrada

- n **eventos** que devem ser agendados em um conjunto de períodos divididos em dias
- r **salas** cada uma com sua capacidade específica
- conjunto com f **recursos** que podem ser disponibilizados em uma sala
- s **estudantes** que desejam assistir um ou mais eventos
- a **disponibilidade** de cada evento - nem todos os eventos podem ser agendados em todos os períodos
- um conjunto de requerimentos de **precedência** que informa que certos eventos devem ser agendados antes de outros

Programação de Cursos Pós-Matrícula - II TTCOMP

Dados de Entrada

- n **eventos** que devem ser agendados em um conjunto de períodos divididos em dias
- r **salas** cada uma com sua capacidade específica
- conjunto com f **recursos** que podem ser disponibilizados em uma sala
- s **estudantes** que desejam assistir um ou mais eventos
- a **disponibilidade** de cada evento - nem todos os eventos podem ser agendados em todos os períodos
- um conjunto de requerimentos de **precedência** que informa que certos eventos devem ser agendados antes de outros

Dados de Entrada

- n **eventos** que devem ser agendados em um conjunto de períodos divididos em dias
- r **salas** cada uma com sua capacidade específica
- conjunto com f **recursos** que podem ser disponibilizados em uma sala
- s **estudantes** que desejam assistir um ou mais eventos
- a **disponibilidade** de cada evento - nem todos os eventos podem ser agendados em todos os períodos
- um conjunto de requerimentos de **precedência** que informa que certos eventos devem ser agendados antes de outros

Dados de Entrada

- n **eventos** que devem ser agendados em um conjunto de períodos divididos em dias
- r **salas** cada uma com sua capacidade específica
- conjunto com f **recursos** que podem ser disponibilizados em uma sala
- s **estudantes** que desejam assistir um ou mais eventos
- a **disponibilidade** de cada evento - nem todos os eventos podem ser agendados em todos os períodos
- um conjunto de requerimentos de **precedência** que informa que certos eventos devem ser agendados antes de outros

Dados de Entrada

- n **eventos** que devem ser agendados em um conjunto de períodos divididos em dias
- r **salas** cada uma com sua capacidade específica
- conjunto com f **recursos** que podem ser disponibilizados em uma sala
- s **estudantes** que desejam assistir um ou mais eventos
- a **disponibilidade** de cada evento - nem todos os eventos podem ser agendados em todos os períodos
- um conjunto de requerimentos de **precedência** que informa que certos eventos devem ser agendados antes de outros

Dados de Entrada

- n **eventos** que devem ser agendados em um conjunto de períodos divididos em dias
- r **salas** cada uma com sua capacidade específica
- conjunto com f **recursos** que podem ser disponibilizados em uma sala
- s **estudantes** que desejam assistir um ou mais eventos
- a **disponibilidade** de cada evento - nem todos os eventos podem ser agendados em todos os períodos
- um conjunto de requerimentos de **precedência** que informa que certos eventos devem ser agendados antes de outros

Programação de Cursos Pós-Matrícula - II TTCOMP

Restrições Fortes

1. cada estudante deve assistir, no máximo, um evento por período
2. a alocação da sala deve respeitar sua capacidade e garantir que a mesma provê os recursos suficientes para o evento
3. cada sala deve receber no máximo um evento por período
4. a disponibilidade dos eventos deve ser respeitada
5. a precedência requerida entre os eventos deve ser respeitada

Restrições Fracas

1. estudantes não devem ser alocados para eventos nos últimos períodos do dia
2. estudantes não devem assistir mais que três eventos por dia
3. as alocações devem evitar que algum estudante tenha apenas um evento agendado em um determinado dia

Programação de Cursos Pós-Matrícula - II TTCOMP

Restrições Fortes

1. cada estudante deve assistir, no máximo, um evento por período
2. a alocação da sala deve respeitar sua capacidade e garantir que a mesma provê os recursos suficientes para o evento
3. cada sala deve receber no máximo um evento por período
4. a disponibilidade dos eventos deve ser respeitada
5. a precedência requerida entre os eventos deve ser respeitada

Restrições Fracas

1. estudantes não devem ser alocados para eventos nos últimos períodos do dia
2. estudantes não devem assistir mais que três eventos por dia
3. as alocações devem evitar que algum estudante tenha apenas um evento agendado em um determinado dia

O Problema de Programação de Exames em Universidades

- resolvido até 4 vezes por ano
- conjunto de alunos que devem realizar um ou mais exames
- restrição forte principal: evitar conflitos para alunos
  - restrição eventualmente relaxada
- restrição fraca principal: espalhar os exames dos alunos
  - especialmente evitar dois exames consecutivos
- recursos limitados tipicamente considerados:
  - salas com capacidade adequada
  - supervisores

O Problema de Programação de Exames em Universidades

- resolvido até 4 vezes por ano
- conjunto de alunos que devem realizar um ou mais exames
- restrição forte principal: evitar conflitos para alunos
  - restrição eventualmente relaxada
- restrição fraca principal: espalhar os exames dos alunos
  - especialmente evitar dois exames consecutivos
- recursos limitados tipicamente considerados:
  - salas com capacidade adequada
  - supervisores

O Problema de Programação de Exames em Universidades

- resolvido até 4 vezes por ano
- conjunto de alunos que devem realizar um ou mais exames
- restrição forte principal: evitar conflitos para alunos
  - restrição eventualmente relaxada
- restrição fraca principal: espalhar os exames dos alunos
  - especialmente evitar dois exames consecutivos
- recursos limitados tipicamente considerados:
  - salas com capacidade adequada
  - supervisores

O Problema de Programação de Exames em Universidades

- resolvido até 4 vezes por ano
- conjunto de alunos que devem realizar um ou mais exames
- restrição forte principal: evitar conflitos para alunos
  - restrição eventualmente relaxada
- restrição fraca principal: espalhar os exames dos alunos
  - especialmente evitar dois exames consecutivos
- recursos limitados tipicamente considerados:
  - salas com capacidade adequada
  - supervisores

O Problema de Programação de Exames em Universidades

- resolvido até 4 vezes por ano
- conjunto de alunos que devem realizar um ou mais exames
- restrição forte principal: evitar conflitos para alunos
  - restrição eventualmente relaxada
- restrição fraca principal: espalhar os exames dos alunos
  - especialmente evitar dois exames consecutivos
- recursos limitados tipicamente considerados:
  - salas com capacidade adequada
  - supervisores

O Problema de Programação de Exames em Universidades

- resolvido até 4 vezes por ano
- conjunto de alunos que devem realizar um ou mais exames
- restrição forte principal: evitar conflitos para alunos
  - restrição eventualmente relaxada
- restrição fraca principal: espalhar os exames dos alunos
  - especialmente evitar dois exames consecutivos
- recursos limitados tipicamente considerados:
  - salas com capacidade adequada
  - supervisores

O Problema de Programação de Exames em Universidades

- resolvido até 4 vezes por ano
- conjunto de alunos que devem realizar um ou mais exames
- restrição forte principal: evitar conflitos para alunos
  - restrição eventualmente relaxada
- restrição fraca principal: espalhar os exames dos alunos
  - especialmente evitar dois exames consecutivos
- recursos limitados tipicamente considerados:
  - salas com capacidade adequada
  - supervisores

O Problema de Programação de Exames em Universidades - versão não capacitada

Dados de Entrada

- N: número de exames;
- E_i: exame, com $i \in \{1, \ldots, N\}$
- B: conjunto de todos os N exames, com $B = \{E_1, \ldots, E_N\}$
- D: conjunto de dias;
- T: número de períodos disponíveis (em todos os dias);
- M: número de estudantes;
- $C_{N \times N}$: matriz de conflitos onde c_{ij} indica quantos alunos devem realizar tanto o exame i quanto o exame j;

Variáveis

$$t_k, k \in \{1, \ldots, N\}, t_k \in \{1, \ldots, T\}$$

O Problema de Programação de Exames em Universidades - versão não capacitada

Dados de Entrada

- N: número de exames;
- E_i: exame, com $i \in \{1, \ldots, N\}$
- B: conjunto de todos os N exames, com $B = \{E_1, \ldots, E_N\}$
- D: conjunto de dias;
- T: número de períodos disponíveis (em todos os dias);
- M: número de estudantes;
- $C_{N \times N}$: matriz de conflitos onde c_{ij} indica quantos alunos devem realizar tanto o exame i quanto o exame j;

Variáveis

$$t_k, k \in \{1, \ldots, N\}, t_k \in \{1, \ldots, T\}$$

O Problema de Programação de Exames em Universidades - versão não capacitada

$$minimizar: \quad \frac{\sum_{i=1}^{N-1} F_1(i)}{M}$$

$$sendo\ que:$$

$$F_1(i) = \sum_{j=i+1}^{N} c_{ij} \cdot proximidade(t_i, t_j)$$

$$proximidade(t_i, t_j) = \begin{cases} 2^5/2^{|t_i - t_j|} & se\ |t_i - t_j| \leq 5 \\ 0 & caso\ contrário \end{cases}$$

$$sujeito\ a:$$

$$\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} c_{ij} \cdot \lambda(t_i, t_j) = 0$$

$$sendo\ que:$$

$$\lambda(t_i, t_j) = \begin{cases} 1 & se\ t_i = t_j \\ 0 & caso\ contrário \end{cases}$$

O Problema de Programação de Exames em Universidades - versão não capacitada

$$minimizar: \quad \frac{\sum_{i=1}^{N-1} F_1(i)}{M}$$

$$sendo\ que:$$

$$F_1(i) = \sum_{j=i+1}^{N} c_{ij} \cdot proximidade(t_i, t_j)$$

$$proximidade(t_i, t_j) = \begin{cases} 2^5/2^{|t_i - t_j|} & se\ |t_i - t_j| \leq 5 \\ 0 & caso\ contrário \end{cases}$$

$$sujeito\ a:$$

$$\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} c_{ij} \cdot \lambda(t_i, t_j) = 0$$

$$sendo\ que:$$

$$\lambda(t_i, t_j) = \begin{cases} 1 & se\ t_i = t_j \\ 0 & caso\ contrário \end{cases}$$

O Problema de Programação de Exames em Universidades - versão não capacitada

$$minimizar: \quad \frac{\sum_{i=1}^{N-1} F_1(i)}{M}$$

sendo que:

$$F_1(i) = \sum_{j=i+1}^{N} c_{ij} \cdot proximidade(t_i, t_j)$$

$$proximidade(t_i, t_j) = \begin{cases} 2^5 / 2^{|t_i - t_j|} & se \ |t_i - t_j| \leq 5 \\ 0 & caso \ contrário \end{cases}$$

sujeito a:

$$\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} c_{ij} \cdot \lambda(t_i, t_j) = 0$$

sendo que:

$$\lambda(t_i, t_j) = \begin{cases} 1 & se \ t_i = t_j \\ 0 & caso \ contrário \end{cases}$$

O Problema de Programação de Exames em Universidades - versão não capacitada

$$minimizar: \quad \frac{\sum_{i=1}^{N-1} F_1(i)}{M}$$

$$sendo\ que:$$

$$F_1(i) = \sum_{j=i+1}^{N} c_{ij} \cdot proximidade(t_i, t_j)$$

$$proximidade(t_i, t_j) = \begin{cases} 2^5/2^{|t_i - t_j|} & se\ |t_i - t_j| \leq 5 \\ 0 & caso\ contrário \end{cases}$$

$$sujeito\ a:$$

$$\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} c_{ij} \cdot \lambda(t_i, t_j) = 0$$

$$sendo\ que:$$

$$\lambda(t_i, t_j) = \begin{cases} 1 & se\ t_i = t_j \\ 0 & caso\ contrário \end{cases}$$

O Problema de Programação de Exames em Universidades - versão não capacitada

O Problema de Programação de Exames em Universidades

Leitura Recomendada

- Carter, M.W. **A Survey of Practical Applications of Examination Timetabling Algorithms**. Operations Research, 34, 1986.
- Carter, M.W. e Laporte, G. **Recent Developments in Practical Examination Timetabling**. In Edmund Burke and Peter Ross, editors, The Practice and Theory of Automated Timetabling. Lecture Notes in Computer Science 1153, 3-21. Springer-Verlag, Berlin, 1996.

O Problema de Alocação de Salas

- Decomposição do problema anterior

Restrições Fortes

- evitar conflitos
- respeitar a capacidade das salas
- respeitar pré-alocações

O Problema de Alocação de Salas

Restrições Fracas

- garantir a alocação de salas com os recursos necessários para cada atividade
- evitar a utilização "salas especiais"
- sempre que possível, manter na mesma sala alunos de um mesmo curso
- sempre que possível, alocar todas as aulas semanais de uma turma de uma disciplina a uma mesma sala
- evitar alocar aulas de turmas pequenas a salas de maior capacidade
- sempre que possível, deixar diariamente um horário vazio em cada sala, para possibilitar sua limpeza

O Problema de Alocação de Salas

Restrições Fracas

- garantir a alocação de salas com os recursos necessários para cada atividade
- evitar a utilização "salas especiais"
- sempre que possível, manter na mesma sala alunos de um mesmo curso
- sempre que possível, alocar todas as aulas semanais de uma turma de uma disciplina a uma mesma sala
- evitar alocar aulas de turmas pequenas a salas de maior capacidade
- sempre que possível, deixar diariamente um horário vazio em cada sala, para possibilitar sua limpeza

O Problema de Alocação de Salas

Restrições Fracas

- garantir a alocação de salas com os recursos necessários para cada atividade
- evitar a utilização "salas especiais"
- sempre que possível, manter na mesma sala alunos de um mesmo curso
- sempre que possível, alocar todas as aulas semanais de uma turma de uma disciplina a uma mesma sala
- evitar alocar aulas de turmas pequenas a salas de maior capacidade
- sempre que possível, deixar diariamente um horário vazio em cada sala, para possibilitar sua limpeza

O Problema de Alocação de Salas

Restrições Fracas

- garantir a alocação de salas com os recursos necessários para cada atividade
- evitar a utilização "salas especiais"
- sempre que possível, manter na mesma sala alunos de um mesmo curso
- sempre que possível, alocar todas as aulas semanais de uma turma de uma disciplina a uma mesma sala
- evitar alocar aulas de turmas pequenas a salas de maior capacidade
- sempre que possível, deixar diariamente um horário vazio em cada sala, para possibilitar sua limpeza

O Problema de Alocação de Salas

Restrições Fracas

- garantir a alocação de salas com os recursos necessários para cada atividade
- evitar a utilização "salas especiais"
- sempre que possível, manter na mesma sala alunos de um mesmo curso
- sempre que possível, alocar todas as aulas semanais de uma turma de uma disciplina a uma mesma sala
- evitar alocar aulas de turmas pequenas a salas de maior capacidade
- sempre que possível, deixar diariamente um horário vazio em cada sala, para possibilitar sua limpeza

O Problema de Alocação de Salas

Restrições Fracas

- garantir a alocação de salas com os recursos necessários para cada atividade
- evitar a utilização "salas especiais"
- sempre que possível, manter na mesma sala alunos de um mesmo curso
- sempre que possível, alocar todas as aulas semanais de uma turma de uma disciplina a uma mesma sala
- evitar alocar aulas de turmas pequenas a salas de maior capacidade
- sempre que possível, deixar diariamente um horário vazio em cada sala, para possibilitar sua limpeza

O Problema de Alocação de Salas

Leitura Recomendada

- M.W. Carter and C.A. Tovey. **When is the classroom assignment problem hard?** Operations Research Supplement 1, 40, 1992.
- M. J. F. Souza, A. X. Martins, and C. R. Araújo. **Experiências com a utilização de simulated annealing e busca tabu na resolução do problema de alocação de salas**. In Anais do XXXIV Simpósio Brasileiro de Pesquisa Operacional, 2002.
- A. Subramanian, J. M. F. Medeiros, L. A. F. Cabral, and M. J. F. Souza. **Aplicação da metaheurística busca tabu na resolução do problema de alocação de salas do centro de tecnologia da UFPB**. In Anais do XXVI Encontro Nacional de Engenharia de Produção, 2006.

Programação de Horários em Escolas

Turma \ Dia,Per.	Segunda					
	1° Per.	2° Per.	3° Per.	4° Per.	5° Per.	1° Per.
101 A	Prof. João	Prof. João	Profa. Soraia	Prof. Alberto	Profa. Júlia	Prof. João
101 B	Profa. Soraia	Prof. Alberto	Prof. João	Profa. Júlia	Profa. Regina	Profa. Sorai
201 A	Profa. Júlia	Profa. Júlia	Prof. Carlos	Prof. João	Profa. Soraia	Profa. Regina
201 B	Profa. Regina	Prof. Soraia	Prof. Alberto	Prof. Carlos	Prof. Carlos	Prof. Albert
201 C	Prof. Carlos	Prof. Carlos	Profa. Júlia	Profa. Regina	Prof. João	Prof. Carlos
301 A	Prof. Alberto	Profa. Regina	Profa. Regina	Profa. Soraia	Prof. Alberto	Prof. Celso
301 B	Prof. Celso	Prof. Celso	Prof. João	Prof. Artur	Prof. Artur	Prof. Antônio
301 C	Prof. Artur	Prof. Artur	Prof. Celso	Prof. Celso	Prof. Antônio	Profa. Ma

Programação de Horários em Escolas

	Segunda					
Dia,Per. / Turma	1° Per.	2° Per.	3° Per.	4° Per.	5° Per.	1° Per.
101 A	Prof. João	Prof. João	Profa. Soraia	Prof. Alberto	Profa. Júlia	Prof. João
101 B	Profa. Soraia	Prof. Alberto	**Prof. João**	Profa. Júlia	Profa. Regina	Profa. Sorai
201 A	Profa. Júlia	Profa. Júlia	Prof. Carlos	Prof. João	Profa. Soraia	Profa. Regina
201 B	Profa. Regina	Prof. Soraia	Prof. Alberto	Prof. Carlos	Prof. Carlos	Prof. Alberto
201 C	Prof. Carlos	Prof. Carlos	Profa. Júlia	Profa. Regina	Prof. João	Prof. Carlos
301 A	Prof. Alberto	Profa. Regina	Profa. Regina	Profa. Soraia	Prof. Alberto	Prof. Celso
301 B	Prof. Celso	Prof. Celso	**Prof. João**	Prof. Artur	Prof. Artur	Prof. Antônio
301 C	Prof. Artur	Prof. Artur	Prof. Celso	Prof. Celso	Prof. Antônio	Profa. Ma

O Problema Clássico de Programação de Horários Professores × Turmas - PPT, Gotlieb, 63

Considere:

- D: dias, elementos $1, \ldots, |D|$;
- H: horários diários, elementos $1, \ldots, |H|$;
- P: professores, elementos $1, \ldots, |P|$;
- $\tilde{P}_{|P| \times |D| \times |H|}$, disponibilidade de professores:
  - $\tilde{p}_{pdh} = \begin{cases} 1 & \text{disponível} \\ 0 & \text{indisponível} \end{cases}$
- T: turmas, elementos $1, \ldots, |T|$;
- $\tilde{R}_{|P| \times |T|}$: sendo que $\tilde{r}_{pt}$ indica quantas aulas o professor p deve lecionar para a turma t

O Problema Clássico de Programação de Horários Professores × Turmas - PPT, Gotlieb, 63

Considere:

- D: dias, elementos $1, \ldots, |D|$;
- H: horários diários, elementos $1, \ldots, |H|$;
- P: professores, elementos $1, \ldots, |P|$;
- $\tilde{P}_{|P| \times |D| \times |H|}$, disponibilidade de professores:
  - $\tilde{p}_{pdh} = \begin{cases} 1 & \text{disponível} \\ 0 & \text{indisponível} \end{cases}$
- T: turmas, elementos $1, \ldots, |T|$;
- $\tilde{R}_{|P| \times |T|}$: sendo que $\tilde{r}_{pt}$ indica quantas aulas o professor p deve lecionar para a turma t

O Problema Clássico de Programação de Horários Professores $\times$ Turmas - PPT, Gotlieb, 63

Considere:

- D: dias, elementos $1, \ldots, |D|$;
- H: horários diários, elementos $1, \ldots, |H|$;
- P: professores, elementos $1, \ldots, |P|$;
- $\tilde{P}_{|P| \times |D| \times |H|}$, disponibilidade de professores:
  - $\tilde{p}_{pdh} = \begin{cases} 1 & \text{disponível} \\ 0 & \text{indisponível} \end{cases}$
- T: turmas, elementos $1, \ldots, |T|$;
- $\tilde{R}_{|P| \times |T|}$: sendo que $\tilde{r}_{pt}$ indica quantas aulas o professor p deve lecionar para a turma t

O Problema Clássico de Programação de Horários Professores × Turmas - PPT, Gotlieb, 63

Considere:

- D: dias, elementos $1, \ldots, |D|$;
- H: horários diários, elementos $1, \ldots, |H|$;
- P: professores, elementos $1, \ldots, |P|$;
- $\tilde{P}_{|P| \times |D| \times |H|}$, disponibilidade de professores:
  - $\tilde{p}_{pdh} = \begin{cases} 1 & \text{disponível} \\ 0 & \text{indisponível} \end{cases}$
- T: turmas, elementos $1, \ldots, |T|$;
- $\tilde{R}_{|P| \times |T|}$: sendo que $\tilde{r}_{pt}$ indica quantas aulas o professor p deve lecionar para a turma t

O Problema Clássico de Programação de Horários Professores × Turmas - PPT, Gotlieb, 63

Considere:

- D: dias, elementos $1, \ldots, |D|$;
- H: horários diários, elementos $1, \ldots, |H|$;
- P: professores, elementos $1, \ldots, |P|$;
- $\tilde{P}_{|P|\times|D|\times|H|}$, disponibilidade de professores:
  - $\tilde{p}_{pdh} = \begin{cases} 1 & \text{disponível} \\ 0 & \text{indisponível} \end{cases}$
- T: turmas, elementos $1, \ldots, |T|$;
- $\tilde{R}_{|P|\times|T|}$: sendo que $\tilde{r}_{pt}$ indica quantas aulas o professor p deve lecionar para a turma t

O Problema Clássico de Programação de Horários
Professores × Turmas - PPT, Gotlieb, 63

Considere:

- D: dias, elementos $1, \ldots, |D|$;
- H: horários diários, elementos $1, \ldots, |H|$;
- P: professores, elementos $1, \ldots, |P|$;
- $\tilde{P}_{|P| \times |D| \times |H|}$, disponibilidade de professores:
  - $\tilde{p}_{pdh} = \begin{cases} 1 & \text{disponível} \\ 0 & \text{indisponível} \end{cases}$
- T: turmas, elementos $1, \ldots, |T|$;
- $\tilde{R}_{|P| \times |T|}$: sendo que $\tilde{r}_{pt}$ indica quantas aulas o professor p deve lecionar para a turma t

Encontrar:

$$x_{ptdh} = \begin{cases} 1 & \text{professor } p \text{ lecionando para a turma } t \text{ no dia } d \text{ período } h \\ 0 & \text{caso contrário} \end{cases}$$

Sujeito a:

$$\sum_{d \in D} \sum_{h \in H} x_{ptdh} = \tilde{r}_{pt} \quad \forall p \in P, t \in T$$

$$\sum_{p \in P} x_{ptdh} \leq 1 \quad \forall t \in T, d \in D, h \in H$$

$$\sum_{t \in T} x_{ptdh} \leq \tilde{t}_{ikl} \quad \forall p \in P, d \in D, h \in H$$

Encontrar:

$$x_{ptdh} = \begin{cases} 1 & \text{professor } p \text{ lecionando para a turma } t \text{ no dia } d \text{ período } h \\ 0 & \text{caso contrário} \end{cases}$$

Sujeito a:

$$\sum_{d \in D} \sum_{h \in H} x_{ptdh} = \tilde{r}_{pt} \quad \forall p \in P, t \in T$$

$$\sum_{p \in P} x_{ptdh} \leq 1 \quad \forall t \in T, d \in D, h \in H$$

$$\sum_{t \in T} x_{ptdh} \leq \tilde{t}_{ikl} \quad \forall p \in P, d \in D, h \in H$$

O Problema Clássico de Programação de Horários Professores × Turmas - PPT

Encontrar:

$$x_{ptdh} = \begin{cases} 1 & \text{professor } p \text{ lecionando para a turma } t \text{ no dia } d \text{ período } h \\ 0 & \text{caso contrário} \end{cases}$$

Sujeito a:

$$\sum_{d \in D} \sum_{h \in H} x_{ptdh} = \tilde{r}_{pt} \quad \forall p \in P, t \in T$$

$$\sum_{p \in P} x_{ptdh} \leq 1 \quad \forall t \in T, d \in D, h \in H$$

$$\sum_{t \in T} x_{ptdh} \leq \tilde{t}_{ikl} \quad \forall p \in P, d \in D, h \in H$$

O Problema Clássico de Programação de Horários Professores × Turmas - PPT

Encontrar:

$$x_{ptdh} = \begin{cases} 1 & \text{professor } p \text{ lecionando para a turma } t \text{ no dia } d \text{ período } h \\ 0 & \text{caso contrário} \end{cases}$$

Sujeito a:

$$\sum_{d \in D} \sum_{h \in H} x_{ptdh} = \tilde{r}_{pt} \quad \forall p \in P, t \in T$$

$$\sum_{p \in P} x_{ptdh} \leq 1 \quad \forall t \in T, d \in D, h \in H$$

$$\sum_{t \in T} x_{ptdh} \leq \tilde{t}_{ikl} \quad \forall p \in P, d \in D, h \in H$$

Resultados Teóricos

- Problema de Decisão NP-Completo
  - Disponibilidade de Professores
  - Pré-Alocações

Tratando Problemas Reais de Programação de Horários em Escolas

Corpo docente:

- requisições de **compacidade** de horários

Período \ Dia	Segunda	Terça	Quarta	Quinta	Sexta
1	300A				100C
2	300A				100C
3					
4			300A		
5			200B	200B	
Dias com atividade	●		●	●	●

Tratando Problemas Reais de Programação de Horários em Escolas

Corpo docente:

- requisições de **compacidade** de horários

Período \ Dia	Segunda	Terça	Quarta	Quinta	Sexta
1	300A				
2	300A	300A			
3	100C				
4	100C	200B			
5	200B				
Dias com atividade					

Tratando Problemas Reais de Programação de Horários em Escolas

Corpo docente:

- requisições de **compacidade** de horários

Período \ Dia	Segunda	Terça	Quarta	Quinta	Sexta
1	300A				
2	300A	300A			
3	100C	200B			
4	100C				
5	200B				
Dias com atividade	●	●			

Tratando Problemas Reais de Programação de Horários em Escolas

Questões Pedagógicas, Horário das Turmas

- distribuição das aulas no decorrer da semana

Dia / Período	Segunda	Terça	Quarta	Quinta	Sexta
1	**Prof. João**	Prof. Carlos	Profa. Soraia	Profa. Júlia	Prof. Alberto
2	**Prof. João**	Prof. Carlos	Profa. Soraia	Profa. Júlia	Prof. Alberto
3	**Prof. João**	Prof. Carlos	Profa. Soraia	Profa. Júlia	Prof. Alberto
4	**Prof. João**	Prof. Carlos	Profa. Soraia	Profa. Júlia	Profa. Joana
5	**Prof. João**	Profa. Júlia	Prof. Alberto	Prof. Carlos	Profa. Joana

Tratando Problemas Reais de Programação de Horários em Escolas

Questões Pedagógicas, Horário das Turmas

- distribuição das aulas no decorrer da semana

Período \ Dia	Segunda	Terça	Quarta	Quinta	Sexta
1	**Prof. João**	Prof. Alberto	Profa. Júlia	Profa. Júlia	Profa. Soraia
2	**Prof. João**	Prof. Alberto	Profa. Júlia	Profa. Júlia	Profa. Soraia
3	Prof. Carlos	Profa. Soraia	**Prof. João**	Profa. Joana	Prof. João
4	Prof. Carlos	Profa. Soraia	**Prof. João**	Profa. Joana	Prof. Carlos
5	Profa. Júlia	Profa. Joana	Prof. Carlos	Prof. Alberto	Prof. Alberto

O Problema de Programação de Horários Professor × Turma com Compacidade - **PPTC**

Restrições fortes

- Não ocorrência de conflitos
- Respeito a um limite diário de aulas por Professor × Turma
  - $\tilde{m}_{pt}$
- Disponibilidade de professores

Restrições fracas

- Oferecimento de quadros de horários compactos (nr. de dias) para professores
- Não ocorrência de "buracos" na agenda dos professores
- Pedidos de aulas geminadas
  - $\tilde{g}_{pt}$

O Problema de Programação de Horários Professor × Turma com Compacidade - **PPTC**

Restrições fortes

- Não ocorrência de conflitos
- Respeito a um limite diário de aulas por Professor × Turma
  - $\tilde{m}_{pt}$
- Disponibilidade de professores

Restrições fracas

- Oferecimento de quadros de horários compactos (nr. de dias) para professores
- Não ocorrência de "buracos" na agenda dos professores
- Pedidos de aulas geminadas
  - $\tilde{g}_{pt}$

O Problema de Programação de Horários Professor × Turma com Compacidade - **PPTC** - Relevância

Compacidade de Horários

- A. Hertz. Tabu search for large scale timetabling problems. *European Journal or Operational Research*, 54:39–47, 1991.
- R. Alvarez-Valdés, G. Martin, and M. Tamarit. Constructing good solutions for the spanish school timetabling problem. *Journal of the Operational Research Society*, 47, 1996.
- A. Colorni, M. Dorigo, and V. Maniezzo. Metaheuristics for high-school timetabling. *Computational Optimization and Applications*, 9(3), 1998.
- A. Schaerf. Local search techniques for large high school timetabling problems. *IEEE Transactions on Systems, Man and Cybernetics Part A:systems and Humans*, 29(4), 1999.
- M.J.F. Souza, L.S. Ochi, and N. Maculan. *A GRASP-Tabu Search Algorithm for solving School Timetabling Problems*, Kluwer Academic Publishers, 2003.
- ...

O Problema de Programação de Horários Professor × Turma com Compacidade - **PPTC** - Relevância

Duração das aulas

- D. Costa. A tabu search algorithm for computing an operational timetable. *European Journal of Operational Research*, 76, 1994.
- A. Drexl and F. Salewski. Distribution requirements and compactness constraints in school timetabling. *European Journal of Operational Research*, 102(1), 1997.
- A. Colorni, M. Dorigo, and V. Maniezzo. Metaheuristics for high-school timetabling. *Computational Optimization and Applications*, 9(3), 1998.
- A. Schaerf. Local search techniques for large high school timetabling problems. *IEEE Transactions on Systems, Man and Cybernetics Part A:systems and Humans*, 29(4), 1999.
- M.J.F. Souza, L.S. Ochi, and N. Maculan. *A GRASP-Tabu Search Algorithm for solving School Timetabling Problems*, Kluwer Academic Publishers, 2003.
- ...

$$
\begin{aligned}
(P) \quad & \textit{minimize } c^T.x && (1)\\
s.a.:\ Ax &\geq b && (2)\\
x_j &\in \{0,1\} \ \ \forall j \in \mathcal{B} && (3)\\
x_j &\in Z^+ \ \ \forall j \in \mathcal{G} && (4)\\
x_j &\geq 0 \ \ \forall j \in \mathcal{C} && (5)
\end{aligned}
$$

Programação Linear Inteira Mista

Aplicação

- aplicações iniciais consideravam apenas simplificações de PPHs
- renovado interesse, com aplicações que consideram problemas reais

Vantagens

- especificação clara do problema
- existência de resolvedores PLIM de alta qualidade
  - alguns disponíveis livremente
    Linderoth J. T. and T. K. Ralphs. Noncommercial software for mixed-integer linear programming. In J. Karlof, editor, Integer Programming: Theory and Practice, 2005.
- avanços contínuos e significativos:
  - pré-processamento
  - planos de corte
  - seleção de nós
- garantia de obtenção da solução ótima

Programação Linear Inteira Mista

Aplicação

- aplicações iniciais consideravam apenas simplificações de PPHs
- renovado interesse, com aplicações que consideram problemas reais

Vantagens

- especificação clara do problema
- existência de resolvedores PLIM de alta qualidade
  - alguns disponíveis livremente

    Linderoth J. T. and T. K. Ralphs. **Noncommercial software for mixed-integer linear programming**. In J. Karlof, editor, Integer Programming: Theory and Practice, 2005.
- avanços contínuos e significativos:
  - pré-processamento
  - planos de corte
  - seleção de nós
- garantia de obtenção da solução ótima

Programação Linear Inteira Mista

Aplicação

- aplicações iniciais consideravam apenas simplificações de PPHs
- renovado interesse, com aplicações que consideram problemas reais

Vantagens

- especificação clara do problema
- existência de resolvedores PLIM de alta qualidade
  - alguns disponíveis livremente

    Linderoth J. T. and T. K. Ralphs. **Noncommercial software for mixed-integer linear programming**. In J. Karlof, editor, Integer Programming: Theory and Practice, 2005.
- avanços contínuos e significativos:
  - pré-processamento
  - planos de corte
  - seleção de nós
- garantia de obtenção da solução ótima

Programação Linear Inteira Mista

Aplicação

- aplicações iniciais consideravam apenas simplificações de PPHs
- renovado interesse, com aplicações que consideram problemas reais

Vantagens

- especificação clara do problema
- existência de resolvedores PLIM de alta qualidade
  - alguns disponíveis livremente

    Linderoth J. T. and T. K. Ralphs. **Noncommercial software for mixed-integer linear programming**. In J. Karlof, editor, Integer Programming: Theory and Practice, 2005.
- avanços contínuos e significativos:
  - pré-processamento
  - planos de corte
  - seleção de nós
- garantia de obtenção da solução ótima

Desvantagens

- método enumerativo
- tempo potencialmente muito longo para prova da otimalidade ou mesmo para a obtenção de uma solução viável

Leitura Recomendada

Título:	Otimização Linear
Autor:	Nelson Maculan e Márcia H. Costa Fampa
Editora:	UnB
Ano:	2006

Título:	Integer Programming
Autor:	Laurence A. Wolsey
Editora:	Wiley-Interscience
Ano:	1998

Formulação de PLIM $\mathcal{F}_1$ para o PPTC

Função Objetivo

$$\text{Minimizar: } \sum_{p\in P}\sum_{d\in D} w_p^{''} v_{pd} + \sum_{p\in P}\sum_{d\in D} w_p^{'} b_{pd} + \sum_{p\in P}\sum_{t\in T} w_p^{'''} g_{pt}$$

- v_{pd}: professor p tem alguma atividade no dia d ($v_{pd} = 1$, $v_{pd} = 0$ caso contrário)
- b_{pd}: # buracos para o professor p no dia d
- g_{pt}: # aulas duplas não fornecidas para o prof. p e turma t

Função Objetivo

$$\text{Minimizar: } \sum_{p\in P}\sum_{d\in D} w_p^{''} v_{pd} + \sum_{p\in P}\sum_{d\in D} w_p^{'} b_{pd} + \sum_{p\in P}\sum_{t\in T} w_p^{'''} g_{pt}$$

- v_{pd}: professor p tem alguma atividade no dia d ($v_{pd} = 1$, $v_{pd} = 0$ caso contrário)
- b_{pd}: # buracos para o professor p no dia d
- g_{pt}: # aulas duplas não fornecidas para o prof. p e turma t

Função Objetivo

Minimizar: $$\sum_{p \in P} \sum_{d \in D} w_p^{''} v_{pd} + \sum_{p \in P} \sum_{d \in D} w_p^{'} b_{pd} + \sum_{p \in P} \sum_{t \in T} w_p^{'''} g_{pt}$$

- v_{pd}: professor p tem alguma atividade no dia d ($v_{pd} = 1$, $v_{pd} = 0$ caso contrário)
- b_{pd}: # buracos para o professor p no dia d
- g_{pt}: # aulas duplas não fornecidas para o prof. p e turma t

Formulação de PLIM $\mathcal{F}_1$ para o PPTC

Buracos para um dado professor e dia

$$b_{pd} \geq \underline{a}_{pd} - \overline{a}_{pd} + v_{pd} - \sum_{t \in T} \sum_{h \in H} x_{ptdh} \quad \forall p \in P, d \in D$$

	Segunda				
Período / Professor	1° Per.	2º Per.	3º Per.	4º Per.	**5º Per.**
João		102 A			102 A

$$\underline{a}_{pd} \geq l. \sum_{t \in T} x_{ptdh} \quad \forall p \in P, d \in D, h \in H$$

$$\overline{a}_{pd} \leq (|H| + 1) - (|H| + 1 - h) \sum_{t \in T} x_{ptdh} \quad \forall p \in P, d \in D, h \in H$$

Buracos para um dado professor e dia

$$b_{pd} \geq \underline{a}_{pd} - \overline{a}_{pd} + v_{pd} - \sum_{t \in T} \sum_{h \in H} x_{ptdh} \quad \forall p \in P, d \in D$$

	Segunda				
Período / Professor	1º Per.	**2º Per.**	3º Per.	4º Per.	5º Per.
João		102 A			102 A

$$\underline{a}_{pd} \geq l. \sum_{t \in T} x_{ptdh} \quad \forall p \in P, d \in D, h \in H$$

$$\overline{a}_{pd} \leq (|H| + 1) - (|H| + 1 - h) \sum_{t \in T} x_{ptdh} \quad \forall p \in P, d \in D, h \in H$$

Aulas Duplas

$\widehat{G}_{pd}$ conjunto de períodos para o professor p, dia d que podem ser o início de uma aula dupla

	Segunda				
Período / Professor	1º Per.	2º Per.	3º Per.	4º Per.	5º Per.
João	✗		✗	●	

● Possível período inicial de uma aula dupla ✗ Indisponibilidade

$$y_{ptdh} \leq x_{ptdh} \quad \forall p \in P, t \in T, d \in D, h \in \widehat{G}_{pd}$$

$$y_{ptdh} \leq x_{ptdh+1} \quad \forall p \in P, t \in T, d \in D, h \in \widehat{G}_{pd}$$

$$g_{pt} \geq \tilde{g}_{pt} - \sum_{d \in D} \sum_{h \in \widehat{G}_{pd}} y_{ptdh} \quad \forall p \in P, t \in T$$

Formulação de PLIM $\mathcal{F}_1$ para o PPTC

Aulas Duplas

$\widehat{G}_{pd}$ conjunto de períodos para o professor p, dia d que podem ser o início de uma aula dupla

	Segunda				
Período / Professor	1º Per.	2º Per.	3º Per.	4º Per.	5º Per.
João	✗		✗	●	

● Possível período inicial de uma aula dupla ✗ Indisponibilidade

$$y_{ptdh} \leq x_{ptdh} \quad \forall p \in P, t \in T, d \in D, h \in \widehat{G}_{pd}$$

$$y_{ptdh} \leq x_{ptdh+1} \quad \forall p \in P, t \in T, d \in D, h \in \widehat{G}_{pd}$$

$$g_{pt} \geq \tilde{g}_{pt} - \sum_{d \in D} \sum_{h \in \widehat{G}_{pd}} y_{ptdh} \quad \forall p \in P, t \in T$$

Aulas Duplas

$\widehat{G}_{pd}$ conjunto de períodos para o professor p, dia d que podem ser o início de uma aula dupla

Professor \ Período	Segunda				
	1º Per.	2º Per.	3º Per.	4º Per.	5º Per.
João	✗		✗	●	

● Possível período inicial de uma aula dupla   ✗ Indisponibilidade

$$y_{ptdh} \leq x_{ptdh} \quad \forall p \in P, t \in T, d \in D, h \in \widehat{G}_{pd}$$

$$y_{ptdh} \leq x_{ptdh+1} \quad \forall p \in P, t \in T, d \in D, h \in \widehat{G}_{pd}$$

$$g_{pt} \geq \tilde{g}_{pt} - \sum_{d \in D} \sum_{h \in \widehat{G}_{pd}} y_{ptdh} \quad \forall p \in P, t \in T$$

Programação Linear Inteira Mista - Trabalhos Relacionados

- T. Birbas, S. Daskalaki, and E. Housos. Timetabling for greek high schools. *Journal of the Operational Research Society*, 48, 1997.
- P. Avella and I. Vasil'ev. A computational study of a cutting plane algorithm for university course timetabling. *Journal of Scheduling*, 8, 2005.
  - Buracos proibidos

Em ambos os casos o usuário tem que especificar o número de dias de trabalho.

Programação Linear Inteira Mista - Trabalhos Relacionados

- T. Birbas, S. Daskalaki, and E. Housos. Timetabling for greek high schools. *Journal of the Operational Research Society*, 48, 1997.
- P. Avella and I. Vasil'ev. A computational study of a cutting plane algorithm for university course timetabling. *Journal of Scheduling*, 8, 2005.
  - Buracos proibidos

Em ambos os casos o usuário tem que especificar o número de dias de trabalho.

Programação Linear Inteira Mista - Trabalhos Relacionados

- T. Birbas, S. Daskalaki, and E. Housos. Timetabling for greek high schools. *Journal of the Operational Research Society*, 48, 1997.
- P. Avella and I. Vasil'ev. A computational study of a cutting plane algorithm for university course timetabling. *Journal of Scheduling*, 8, 2005.
  - Buracos proibidos

Em ambos os casos o usuário tem que especificar o número de dias de trabalho.

Heurísticas Construtivas

- tentam oferecer rapidamente uma solução factível (ou com poucas infactibilidades) e de qualidade rapidamente
- decidem uma alocação por vez

Princípio Geral:

Alocar as aulas mais urgentes nos períodos mais apropriados.

- em geral, obter uma solução sem conflitos já é bastante difícil!
- obter um quadro de horários sem conflitos consiste em resolver o **Problema de Coloração de Vértices de um Grafo**

Heurísticas Construtivas

- tentam oferecer rapidamente uma solução factível (ou com poucas infactibilidades) e de qualidade rapidamente
- decidem uma alocação por vez

Princípio Geral:

Alocar as aulas mais urgentes nos períodos mais apropriados.

- em geral, obter uma solução sem conflitos já é bastante difícil!
- obter um quadro de horários sem conflitos consiste em resolver o **Problema de Coloração de Vértices de um Grafo**

Heurísticas Construtivas

- tentam oferecer rapidamente uma solução factível (ou com poucas infactibilidades) e de qualidade rapidamente
- decidem uma alocação por vez

Princípio Geral:
Alocar as aulas mais urgentes nos períodos mais apropriados.

- em geral, obter uma solução sem conflitos já é bastante difícil!
- obter um quadro de horários sem conflitos consiste em resolver o **Problema de Coloração de Vértices de um Grafo**

Heurísticas Construtivas

- tentam oferecer rapidamente uma solução factível (ou com poucas infactibilidades) e de qualidade rapidamente
- decidem uma alocação por vez

Princípio Geral:
Alocar as aulas mais urgentes nos períodos mais apropriados.

- em geral, obter uma solução sem conflitos já é bastante difícil!
- obter um quadro de horários sem conflitos consiste em resolver o **Problema de Coloração de Vértices de um Grafo**

Heurísticas Construtivas

- tentam oferecer rapidamente uma solução factível (ou com poucas infactibilidades) e de qualidade rapidamente
- decidem uma alocação por vez

Princípio Geral:
Alocar as aulas mais urgentes nos períodos mais apropriados.

- em geral, obter uma solução sem conflitos já é bastante difícil!
- obter um quadro de horários sem conflitos consiste em resolver o **Problema de Coloração de Vértices de um Grafo**

Heurísticas Construtivas

Maior Grau Primeiro

- nós com maior grau são alocados primeiro
- seleciona-se a primeira cor que não cause conflitos

Heurísticas Construtivas

Maior Grau Primeiro - Preenchimento do Topo

- lista ordenada pelo grau dos vértices
- a cada iteração, percorre-se a lista alocando tantos nós quanto possível na primeira cor disponível
- volta-se para o início da lista e continua-se na cor seguinte

Heurísticas Construtivas

Maior Grau Modificado Primeiro

- Prioridade de um vértice i:

$$p_1(i) = \sum_{j \in N(i)} d(j)$$

$N(i)$: vizinhos do vértice i
$d(j)$ grau do vértice j

$$p_{k+1}(i) = \sum_{j \in N(i)} p_k(j)$$

Melhores resultados que "maior grau primeiro".

Maior Grau Modificado Primeiro

- Prioridade de um vértice i:

$$p_1(i) = \sum_{j \in N(i)} d(j)$$

$N(i)$: vizinhos do vértice i
$d(j)$ grau do vértice j

$$p_{k+1}(i) = \sum_{j \in N(i)} p_k(j)$$

Melhores resultados que "maior grau primeiro".

Maior Grau Modificado Primeiro

- Prioridade de um vértice i:

$$p_1(i) = \sum_{j \in N(i)} d(j)$$

$N(i)$: vizinhos do vértice i
$d(j)$ grau do vértice j

$$p_{k+1}(i) = \sum_{j \in N(i)} p_k(j)$$

Melhores resultados que "maior grau primeiro".

Heurísticas Construtivas

Menor Grau Primeiro Recursivo com Troca

- utiliza-se alguma ordenação do tipo "maior grau primeiro"
- retira-se o vértice i da topo da lista, associando-o com a primeira cor onde não ocorrerão conflitos
- se o vértice i conflitar com todas as cores existentes, procura-se uma cor c_j na qual exista apenas um nó j conflitando
  - se possível, mude a cor do vértice j
  - caso contrário, tentar troca bicromática do seguinte modo:

Menor Grau Primeiro Recursivo com Troca

- utiliza-se alguma ordenação do tipo "maior grau primeiro"
- retira-se o vértice i da topo da lista, associando-o com a primeira cor onde não ocorrerão conflitos
- se o vértice i conflitar com todas as cores existentes, procura-se uma cor c_j na qual exista apenas um nó j conflitando
  - se possível, mude a cor do vértice j
  - caso contrário, tentar troca bicromática do seguinte modo:

Menor Grau Primeiro Recursivo com Troca

- utiliza-se alguma ordenação do tipo "maior grau primeiro"
- retira-se o vértice i da topo da lista, associando-o com a primeira cor onde não ocorrerão conflitos
- se o vértice i conflitar com todas as cores existentes, procura-se uma cor c_j na qual exista apenas um nó j conflitando
  - se possível, mude a cor do vértice j
  - caso contrário, tentar troca bicromática do seguinte modo:
    - para cada cor r considere o conjunto C_r de vértices com essa cor, se nenhum vértice em C_r conflitar com i então mude a cor de j para a cor r e a cor dos vértices em C_r para c_j; se a troca não for possível cria-se uma nova cor e continua-se o processo

Heurísticas Construtivas

Menor Grau Primeiro Recursivo com Troca

- utiliza-se alguma ordenação do tipo “maior grau primeiro”
- retira-se o vértice i da topo da lista, associando-o com a primeira cor onde não ocorrerão conflitos
- se o vértice i conflitar com todas as cores existentes, procura-se uma cor c_j na qual exista apenas um nó j conflitando
  - se possível, mude a cor do vértice j
  - caso contrário, tentar troca bicromática do seguinte modo:
    - para cada cor r considere o conjunto C_r de vértices com essa cor, se nenhum vértice em C_r conflitar com i então mude a cor de j para a cor r e a cor dos vértices em C_r para c_j; se a troca não for possível cria-se uma nova cor e continua-se o processo

Heurísticas Construtivas

Menor Grau Primeiro Recursivo com Troca

- utiliza-se alguma ordenação do tipo "maior grau primeiro"
- retira-se o vértice i da topo da lista, associando-o com a primeira cor onde não ocorrerão conflitos
- se o vértice i conflitar com todas as cores existentes, procura-se uma cor c_j na qual exista apenas um nó j conflitando
  - se possível, mude a cor do vértice j
  - caso contrário, tentar troca bicromática do seguinte modo:
    - para cada cor r considere o conjunto C_r de vértices com essa cor, se nenhum vértice em C_r conflitar com i então mude a cor de j para a cor r e a cor dos vértices em C_r para c_j; se a troca não for possível cria-se uma nova cor e continua-se o processo

Heurísticas Construtivas

Vantagem

- rapidez
- pouco consumo de memória
    - Nottingham, final da década de 60

"...a aquisição de um novo computador mais poderoso permitiu o agendamento de 651 exames, utilizando 12 kilobytes de RAM..."

Desvantagens

- geram somente uma solução
- falham com freqüência na produção de quadros de horários sem conflitos

Heurísticas Construtivas

Vantagem

- rapidez
- pouco consumo de memória
  - Nottingham, final da década de 60

"...a aquisição de um novo computador mais poderoso permitiu o agendamento de 651 exames, utilizando 12 kilobytes de RAM..."

Desvantagens

- geram somente uma solução
- falham com freqüência na produção de quadros de horários sem conflitos

Heurísticas Construtivas

Vantagem

- rapidez
- pouco consumo de memória
  - Nottingham, final da década de 60

"...a aquisição de um novo computador mais poderoso permitiu o agendamento de 651 exames, utilizando 12 kilobytes de RAM..."

Desvantagens

- geram somente uma solução
- falham com freqüência na produção de quadros de horários sem conflitos

Busca Local

- dada uma solução inicial, pode-se tentar melhorá-la
  - se a mesma é <u>infactível</u>, irá tentar-se diminuir o número de infactibilidades
  - se a mesma é <u>factível</u>, irá tentar-se melhorar o atendimento das restrições fracas
- um <u>movimento</u> é uma modificação de um tipo em uma solução
  - ex.: modificar o período de alocação de uma aula de um professor
- a vizinhança $\mathcal{N}(s)$ de uma solução s consiste no conjunto de soluções que podem ser alcançadas com a realização de um movimento em s

Busca Local

- dada uma solução inicial, pode-se tentar melhorá-la
  - se a mesma é infactível, irá tentar-se diminuir o número de infactibilidades
  - se a mesma é factível, irá tentar-se melhorar o atendimento das restrições fracas
- um movimento é uma modificação de um tipo em uma solução
  - ex.: modificar o período de alocação de uma aula de um professor
- a vizinhança $\mathcal{N}(s)$ de uma solução s consiste no conjunto de soluções que podem ser alcançadas com a realização de um movimento em s

Busca Local

- dada uma solução inicial, pode-se tentar melhorá-la
  - se a mesma é infactível, irá tentar-se diminuir o número de infactibilidades
  - se a mesma é factível, irá tentar-se melhorar o atendimento das restrições fracas
- um movimento é uma modificação de um tipo em uma solução
  - ex.: modificar o período de alocação de uma aula de um professor
- a vizinhança $\mathcal{N}(s)$ de uma solução s consiste no conjunto de soluções que podem ser alcançadas com a realização de um movimento em s

Busca Local

- dada uma solução inicial, pode-se tentar melhorá-la
  - se a mesma é infactível, irá tentar-se diminuir o número de infactibilidades
  - se a mesma é factível, irá tentar-se melhorar o atendimento das restrições fracas
- um movimento é uma modificação de um tipo em uma solução
  - ex.: modificar o período de alocação de uma aula de um professor
- a vizinhança $\mathcal{N}(s)$ de uma solução s consiste no conjunto de soluções que podem ser alcançadas com a realização de um movimento em s

Busca Local

A Vizinhança Ideal

- baixo esforço computacional para pesquisá-la
- contém soluções melhores, sempre que a solução não for ótima

Busca Local

Professor\Período	Dia 1					Dia 2					Dia 3					Dia 4					Dia 5				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	1	3	2	3	2	3	2	3	1	1	×	×	×	×	×	3	1	2	2	1					
2	2	1	3	2	3	×	×	×	×	×						1	2	3	1	3	2	1	2	1	3
3									2	2	1	1	2	3	3	×	×	×	×	×	1	3	1	3	2
4	3	2	1										3	1	2	×	×	×	×	×					
5	×	×	×	×	×	2	1	7	3	8	3	2	1	7	6						3	2	6	8	1
6	4	8	7	1	5	1	4	6			5	8	7	6	1						×	×	×	×	×
7						6	5	1	7	3	2	4	8			×	×	×	×	×					
8					1	8	3	2	4	5	×	×	×	×	×	6	7	1	3	2					
9	×	×	×	×	×						4	3		2	5	2	3				5	4	3	2	
10				5	4	×	×	×	×	×						5		4	4	5	4	5	5		4
11	5	4	5	4							×	×	×	×	×		4	5	5	4				4	5
12						7	8	5	6	4	7	6	5	8	4	×	×	×	×	×	7	6	4	5	8
13	7	6	6	8	7						×	×	×	×	×	7	6	8	6	8	6	8	8	7	7
14	8	7	8	6	6	×	×	×	×	×						8	8	7	7	6	8	7	7	6	6
15	6	5	4	7	8	4	6	8	5	7	8	5	6	4	7	×	×	×	×	×					
16						5	7	4	8	6	6	7	4	5	8	4	5	6	8	7	×	×	×	×	×

- descobre cadeias de movimentos que se realizados em conjunto melhoram um quadro de horários
- baseado em caminhos mínimos
- SOUZA, M.; OCHI, L.; MACULAN, N. **A GRASP-tabu search algorithm for solving school timetabling problems**. In: Metaheuristics: Computer Decision-Making. : Kluwer Academic Publishers, 2003. p. 659-672

O Procedimento Interturmas-Intraturmas

- descobre cadeias de movimentos que se realizados em conjunto melhoram um quadro de horários
- baseado em caminhos mínimos
- SOUZA, M.; OCHI, L.; MACULAN, N. **A GRASP-tabu search algorithm for solving school timetabling problems**. In: Metaheuristics: Computer Decision-Making. : Kluwer Academic Publishers, 2003. p. 659-672

O Procedimento Interturmas-Intraturmas

- descobre cadeias de movimentos que se realizados em conjunto melhoram um quadro de horários
- baseado em caminhos mínimos
- SOUZA, M.; OCHI, L.; MACULAN, N. **A GRASP-tabu search algorithm for solving school timetabling problems**. In: Metaheuristics: Computer Decision-Making. : Kluwer Academic Publishers, 2003. p. 659-672

O Procedimento Intraturmas-Interturmas

Definições

Inviabilidade do tipo 1: existência de conflitos;

Inviabilidade do tipo 2: excesso de aulas prof. $\times$ turma em um dia.

Representação da Solução

- $Q_{|P| \times (|D|.|P|)}$
- elementos $q_{ph}, p \in P, h \in \{|D| \times |P|\}$

$$q_{ph} = \begin{cases} -1 & \text{professor indisponível} \\ 0 & \text{professor disponível} \\ \{1, \ldots, |T|\} & \text{alocado para uma turma} \end{cases}$$

O Procedimento Intraturmas-Interturmas

Definições

Inviabilidade do tipo 1: existência de conflitos;

Inviabilidade do tipo 2: excesso de aulas prof. × turma em um dia.

Representação da Solução

- $Q_{|P| \times (|D|.|P|)}$
- elementos $q_{ph}, p \in P, h \in \{|D| \times |P|\}$

$$q_{ph} = \begin{cases} -1 & \text{professor indisponível} \\ 0 & \text{professor disponível} \\ \{1, \ldots, |T|\} & \text{alocado para uma turma} \end{cases}$$

Função Objetivo

$$f(Q) \quad = \quad \underbrace{\omega \times f_1(Q)}_{\text{Conflitos}} \quad + \quad \delta \times f_2(Q) \quad + \quad \rho \times f_3(Q)$$

Função Objetivo

$$f(Q) \quad = \quad \omega \times f_1(Q) \quad + \quad \delta \times f_2(Q) \quad + \quad \rho \times f_3(Q)$$

$\uparrow$

Excesso de aulas

Função Objetivo

$$f(Q) \quad = \quad \omega \times f_1(Q) \quad + \quad \delta \times f_2(Q) \quad + \quad \rho \times f_3(Q)$$

$\uparrow$

Restrições fracas

Ativação

- quando $f_1(Q) = 0$
- inicialmente procura-se minimizar $f_2(Q)$
  - $f_2(Q) = 0$, tenta-se minimizar $f_3(Q)$

O Procedimento *Intraturmas*

Exemplo

- $f_1(Q) = f_2(Q) = 0$
- Turma j

Considere

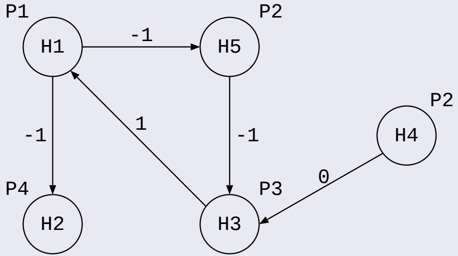
- Grafo $G_j = (V_j, A_j)$
- V_j conjunto de períodos em que a turma j pode ter aula (todos).
- A_j conjunto de arcos com pesos, definidos como segue:
- $A_j = \{(k, \overline{k})$: o professor que leciona para a turma j no horário k está disponível no horário $\overline{k}$ e a restrição forte (e) não está violada nesse horário$\}$.
- custo do arco $(k, \overline{k})$: $c_{k,\overline{k}}$

Exemplo

Horário

	H1	H2	H3	H4	H5		f_i
P1	A		B	B			1
P2	B	C		A	A		1
P3		B	A	C	B		0
P4	C	A	C	D	–		0

Grafo - Turma A



Exemplo

Horário

	H1	H2	H3	H4	H5		f_i
P1	A		B	B			1
P2	B	C		A	A		1
P3		B	A	C	B		0
P4	C	A	C	D	–		0

Quadro de Horários Depois dos Movimento

	H1	H2	H3	H4	H5		f_i
P1			B	B	A		0
P2	B	C	A	A			0
P3	A	B		C	B		1
P4	C	A	C	D	-		0

O Procedimento Intraturmas

Considerações

- procura-se por alguma melhora em cada uma das turmas repetindo enquanto houver sucesso;
- o ciclo de custo negativo encontrado sempre corresponde à uma melhora na f.o.?

O Procedimento Intraturmas

Considerações

- procura-se por alguma melhora em cada uma das turmas repetindo enquanto houver sucesso;
- o ciclo de custo negativo encontrado sempre corresponde à uma melhora na f.o.?

Exemplo

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10		f_i
P1	A		A			A						5
P2		A						B	B	A		4
P3				A	A		A					4
P4								A	A			2

Grafo G_A

Exemplo

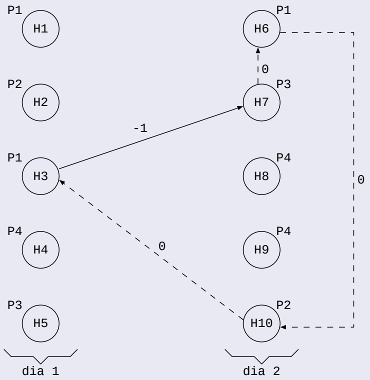
	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10		f_i
P1	A		A			A						5
P2		A						B	B	A		4
P3				A	A		A					4
P4								A	A			2

Grafo G_A

Exemplo

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10		f_i
P1	A		A			A						5
P2		A						B	B	A		4
P3				A	A		A					4
P4								A	A			2

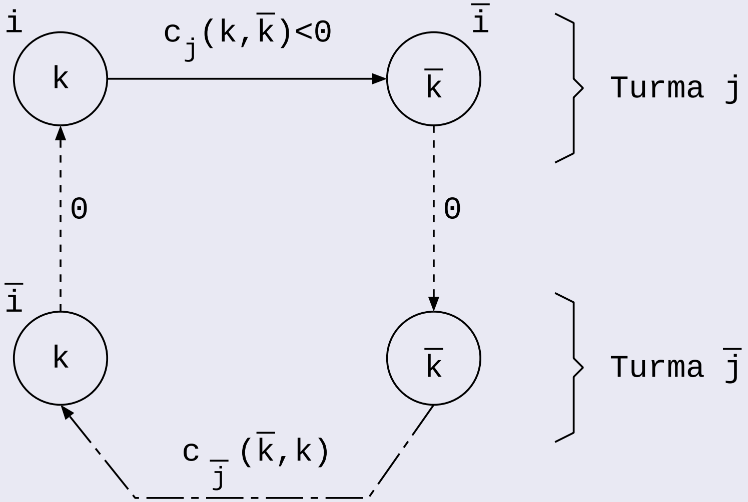
Grafo G_A



Dificuldades

- Falsas seqüências de melhoramento
  - Professor envolvido em mais de um arco do ciclo
- Contornando as dificuldades (heuristicamente)
  - escolhe-se um arco do ciclo descoberto, digamos $(k, \bar{k})$;
  - insere-se em uma lista de arcos proibidos L;
  - procura-se outro ciclo de custo negativo;
  - sucesso ? zera-se L e continua-se o processo;
  - falha ? zera-se L e procede-se para a próxima turma.

Exemplo



O Procedimento Interturmas

Funcionamento geral

- para cada turma j
- para cada arco $(k, \overline{k}) \in A_j$ de custo negativo
  - seja $\overline{i}$ o professor envolvido no horário $\overline{k}$;
  - seja $\overline{j}$ a turma na qual $\overline{i}$ está envolvido no horário $\overline{k}$;
  - se $c_j(k, \overline{k}) + c_{\overline{j}}(\overline{k}, k) < 0$ e f.o. melhorar mantendo a fact.;
    - atualize os quadros de horários de j e $\overline{j}$, bem como G_j e $G_{\overline{j}}$;
    - retorne ao procedimento Intraturmas;